递归式求解

——主方法

The Master Method (通用法，万能法)

- 可迅速求解
  - $T(n) = aT(n/b) + f(n)$ //常数$a \geq 1$，$b > 1$，$f(n)$渐近正
  - 物理意义：将Size为n的问题划分为a个子问题，每个子问题Size为n/b。每个子问题的时间为$T(n/b)$，划分和combine的时间为$f(n)$。
  - Note: n/b不一定为整数，应为$\lfloor n/b \rfloor$或$\lceil n/b \rceil$,不会影响渐近界。
- **Theorem. 1 (Master Theorem，*Page 53*)**

  设$a \geq 1$，$b > 1$是整数，$f(n)$是函数，$T(n)$是定义在非负整数上的递归方程$T(n) = aT(n/b) + f(n)$，这里n/b解释为$\lfloor n/b \rfloor$或$\lceil n/b \rceil$，则$T(n)$的渐近界为：

The Master Method (通用法，万能法)(续)

Master Theorem

- 若$f(n) = O(n^{\log_b a-\epsilon})$对某一常数$\epsilon > 0$成立，则$T(n) = \Theta(n^{\log_b a})$
- 若$f(n) = \Theta(n^{\log_b a})$则$T(n) = \Theta(n^{\log_b a} \cdot \lg n)$
- 若$f(n) = \Omega(n^{\log_b a+\epsilon})$对某一常数$\epsilon > 0$成立，且$af(n/b) \leq cf(n)$对某常数$c < 1$及足够大的$n$成立，则$T(n) = \Theta(f(n))$

证明从略

The Master Method (通用法，万能法)(续)

该定理意义

- 比较$f(n)$和$n^{\log_b a}$，直观上<u>两函数中较大者决定方程的解</u>。

$case\ 1$: $n^{\log_b a}$较大，$\therefore T(n)=\Theta(n^{\log_b a})$

比$f(n)$大一个多项式因子n^{ϵ}

$case\ 3$: $f(n)$较大，$\therefore T(n)=\Theta(f(n))$

比$n^{\log_b a}$大一个多项式因子n^{ϵ}

$case\ 2$: 二者相同，其解乘上一对数因子

$$\therefore T(n)=\Theta\left(n^{\log_b a}\cdot \lg n\right)=\Theta(f(n)\lg n)$$

The Master Method (通用法，万能法)(续)

- **Note：case1 & 3中，比较 f 和$n^{\log_b a}$的大小均是相对多项式因子n^{ϵ}而言**
- ***这三种情况并未覆盖所有可能的 $f(n)$*，即case 1 & 2及case 2 & 3间有间隙。**
- **例1：** $T(n) = 9T(n/3) + n$

  解： $a = 9$，$b = 3$，$f(n) = n$

  $n^{\log_b a} = n^{\log_3 9} = \Theta(n^2)$

  $\therefore f(n) = O(n^{\log_3 9-\epsilon})$，这里$\epsilon = 1$

  故$T(n) = \Theta(n^2)$ $//case1$

The Master Method (通用法，万能法)(续)

- **例2：** $T(n) = T(2n/3) + 1$

  解： $a = 1$，$b = 3/2$，$f(n) = \Theta(1)$

  $n^{\log_b a} = n^{\log_{3/2} 1} = n^0 = 1 \quad //case2$

  $\therefore T(n) = \Theta(\lg n)$

The Master Method (通用法，万能法)(续)

- **例3：** $T(n) = 3T(n/4) + nlgn$

解： $n^{\log_b a} = n^{\log_4 3} = O(n^{0.793})$

$f(n) = n\lg n$

$\because f(n) = \Omega\left(n^{\log_4 3+\epsilon}\right)$

即$f(n)$比$n^{\log_b a}$大一多项式因子$n^{0.2}$

对足够大的n：$af(n/b) = 3(n/4)\lg(n/4)$

$\leq \frac{3}{4} n\lg n = cf(n)$成立

$\therefore$满足$case3$，解为$T(n) = \Theta(n\lg n)$

The Master Method (通用法，万能法)(续)

- **例4：** $T(n) = 2T(n/2) + n\lg n$

解： $n^{\log_b a} = n < f(n) = n\lg n$

但是$f(n)$并不大于n一个多项式因子$n^{\epsilon}(\epsilon > 0)$

$\because$对给定$\epsilon > 0$，对足够大的n，$\lg n < n^{\epsilon}$

$$\frac{n^{\epsilon}}{\lg n} \to \infty$$

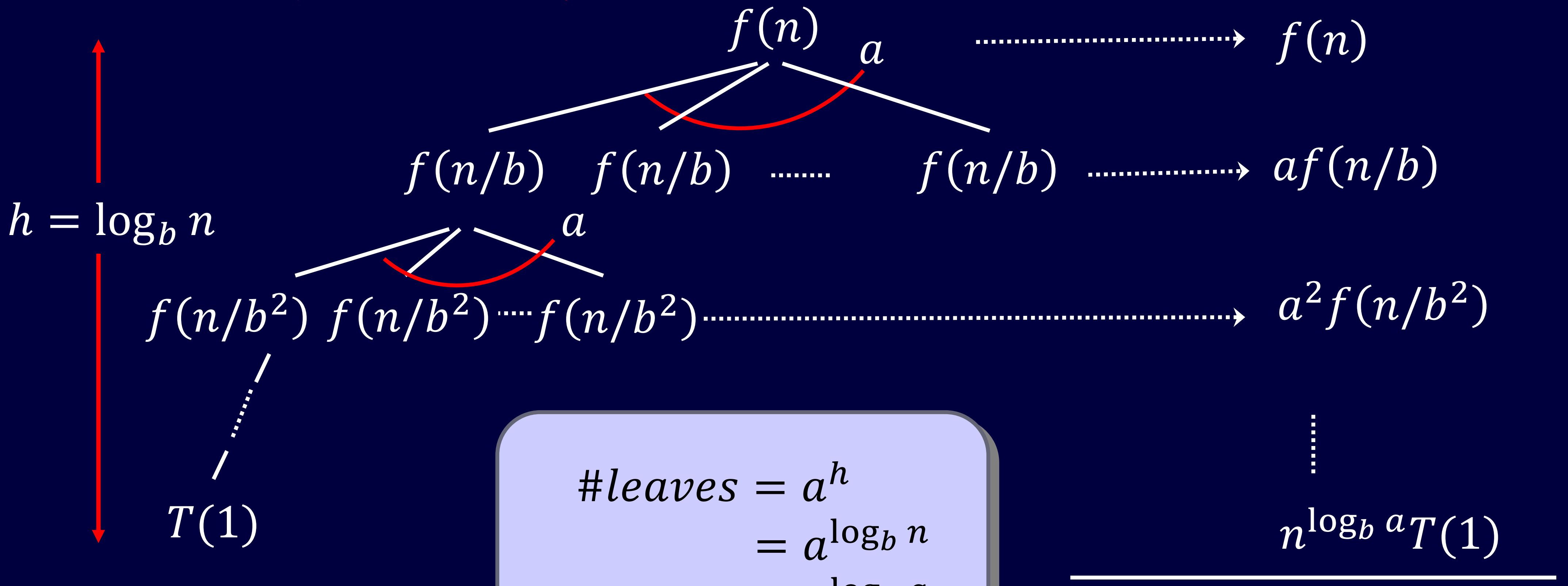
$\therefore$此解属于case2和case3之间

不能用master定理

The Master Method (通用法，万能法)(续)

Idea of master theorem

Recursion tree:



$$
\begin{aligned}
\#leaves &= a^h \\
&= a^{\log_b n} \\
&= n^{\log_b a}
\end{aligned}
$$

Total: $\sum_{k=0}^{\log_b n - 1} a^k f(n/b^k) + \Theta(n^{\log_b a})$